

AD-A150 051

NAVIGATION FOR AN INTELLIGENT MOBILE ROBOT(U)
CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST
J L CROWLEY AUG 84 CMU-RI-TR-84-18

1/1

UNCLASSIFIED

F/G 6/4

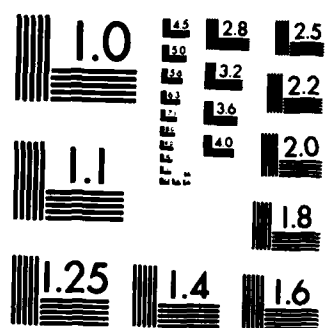
NL



END

FILMED

DTIC



AD-A150 051

12

Navigation for an
Intelligent Mobile Robot

James L. Crowley

CMU-RI-TR-84-18

Carnegie-Mellon University

The Robotics Institute

Technical Report

DTIC

COLLECTED

FEB 07 1985

E

DTIC FILE COPY

85 01 28 145

**Navigation for an
Intelligent Mobile Robot**

James L. Crowley

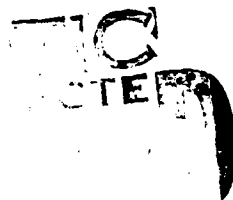
CMU-RI-TR-84-18

**The Laboratory for Household Robots
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213**

August 1984

Copyright © 1984 Carnegie-Mellon University

This research was sponsored by Commodore Business Machines, Inc.; Denning Mobile Robotics, Inc.; and the Commonwealth of Pennsylvania.



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CMU-RI-TR-84-18	2. GOVT ACCESSION NO. AD-A150051	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Navigation for an Intelligent Mobile Robot	5. TYPE OF REPORT & PERIOD COVERED Interim	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) James L. Crowley	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University The Robotics Institute Pittsburgh, PA. 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217	12. REPORT DATE August 1984	13. NUMBER OF PAGES 13
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Approved for public release; distribution unlimited		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper describes a system which performs task-oriented navigation for an intelligent mobile robot. Global path planning in this system is based on a pre-learned model of the robot's domain. The pre-learned model is divided into convex regions using a new "maximal-area" convex decomposition algorithm. A network of convex region entry-points, called "adits," provides the basis for planning a global path as a sequence of straight line movements. This navigation system is based on a dynamically maintained model of the local		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. cont'd.)

environment, called the "Composite Local Model," which integrates information from different sensors and different views, as well as from the pre-learned model of the robot's domain. Local straight line movements are planned and monitored using this local model. The estimated position of the robot is corrected by the difference in position between observed sensor signals and the corresponding symbols in the local model. This system is useful for navigation in a finite, pre-learned domain as a house, office, or factory.

Table of Contents

1 Introduction	1
1.1 The Navigation Problem	1
1.2 Summary of Solution	2
2 Review of Prior Techniques	2
2.1 Find-Path	2
2.2 The Stanford Cart and the C-MU Rover	3
2.3 Hilare	3
2.4 Comment	3
3 The Computational Paradigm	3
3.1 The Composite Local Model	5
3.2 The Sensor Models	5
3.3 Update Local Model	5
4 Navigation	6
4.1 Navigation Modes	6
4.2 The Network of Places and the Global Model	7
4.3 Learning the network of places and the global model	8
4.4 Global Path Planning and Execution	8
4.5 Local Path Execution	10
4.6 Local Path Planning	11
5 Summary and Conclusion	11

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

List of Figures

Figure 1: Framework for an Intelligent Navigation System	4
Figure 2: The Components of the Network of Places	7
Figure 3: An example of a "largest area" convex decomposition of a floor plan.	8
Figure 4: The set of shrunken convex regions and adits that result from the convex regions in figure 3. Adits are shown as boxes.	9
Figure 5: State Transition Diagram for Local Navigation	9
Figure 6: Legal Highway for the Path Execution	10

Abstract

This paper describes a system which performs task-oriented navigation for an intelligent mobile robot. Global path planning in this system is based on a pre-learned model of the robot's domain. The pre-learned model is divided into convex regions using a new "maximal-area" convex decomposition algorithm. A network of convex region entry-points, called "adits", provides the basis for planning a global path as a sequence of straight line movements.

This navigation system is based on a dynamically maintained model of the local environment, called the "Composite Local Model", which integrates information from different sensors and different views, as well as from the pre-learned model of the robot's domain. Local straight line movements are planned and monitored using this local model. The estimated position of the robot is corrected by the difference in position between observed sensor signals and the corresponding symbols in the local model. This system is useful for navigation in a finite, pre-learned domain such as a house, office, or factory.

1 Introduction

This paper describes a system for autonomous navigation by an intelligent mobile robot in a known domain. The first section introduces the navigation problem and summarizes our solution. After a review of previous techniques, the computational paradigm for this navigation system is discussed. Navigation is shown to encompass problems of global path planning and execution, of local path planning and execution, and of position estimation. Techniques are presented for dynamic description of the robots environment using a rotating sonar ranging device, and for representing the learned model of the world as a collection of convex regions. A global path planning technique based on convex regions is presented. Techniques are then presented for local path execution and for dynamic modification of the path plan in response to unexpected obstacles.

The techniques described in this paper are part of an effort to develop a low-cost "Intelligent Mobile Platform" or IMP. By the term "intelligent" we mean that the navigation is "task-oriented" and that it is based on dynamically sensing and modeling the external world. The IMP is designed to respond to commands of the form "Go To <place>" where <place> is a pre-learned location in a network of "learned places". The IMP is able to use its network of places to plan a path to <place>. It is then able to use its sensing, modeling and navigation abilities to execute this plan and to modify the plan dynamically in reaction to unexpected events. The IMP is to serve as a foundation for household, business, and factory robots which require intelligent navigation.

1.1 The Navigation Problem

The task of a navigation system is to plan a path to a specified goal and to execute this plan, modifying it as necessary to avoid unexpected obstacles. When a mobile robot is restricted to a finite domain, the navigation problem is simplified. In finite domains such as a house, office, or factory, it is possible for a mobile robot navigation system learn and use a "Global Model" which describes the floor plan of the domain. This Global Model is used to plan a path to a specified goal and to recall the expected local model of the world as this plan is executed.

Mobile robot navigation in a prelearned domain is easily decomposed into a global navigation problem and a local navigation problem. Each of these two levels require both a planning and an execution phase. At the global level, the system plans and executes a sequence of actions in some suitable atomic units, such as a sequence of goal points. At the local level, the system must plan and execute local actions to accomplish each global action in such a way as to accommodate any unforeseen obstacles.

The planning stage must produce a sequence of actions which are both reasonably efficient and have a high probability of getting the robot to the goal without colliding with an obstacle. In the execution stage, the robot must monitor its motions to verify that it is moving according to the plan. It must also monitor the environment to insure that no unexpected obstacle blocks its movements.

1.2 Summary of Solution

The navigation system described below derives its efficiency from several design decisions. Paths are planned and executed as a sequence of straight-line motions. This yields a global process which plans and executes the sequence of straight line movements, while a local process plans and monitors each straight line movement.

Global path planning is based on a pre-learned "network of places". The network of places is composed of doorway, or "adit", points which are connected by straight lines. Free space is divided into "maximum area" convex regions. Adits describe the doorways to these convex regions.

Local straight line movements are planned and monitored based on a dynamically maintained model of the local environment. This "Composite Local Model" is fundamental to position estimation, local path planning, local path execution, learning and other higher level processes.

The network of places is learned in a special "learn mode". Automatic learning greatly simplifies the practical problem of giving the system an accurate model of the world. Restricting learning to a special mode greatly simplifies its implementation.

The world modeling and navigation procedures for the IMP have been implemented and refined using an interactive mobile robot simulation program [3]. These techniques are currently being reimplemented on a prototype of the IMP which contains two 16 bit microprocessors.

2 Review of Prior Techniques

A number of interesting research results have been obtained on problems which are relevant to mobile robot navigation. A quick review of the salient systems provides a picture of the current state of the scientific art.

2.1 Find-Path

Planning a path based on a model is a problem that is fundamental to intelligent control of robot arms as well as mobile robots. Lozano-Pérez has developed a formal version of the general path planning problem. This formalization is referred to as the "find-path" problem [9]. In its most general form, the goal of find-path is to determine a continuous path for an object from an initial location to a goal location without colliding with an obstacle.

Lozano-Pérez provided a mathematical treatment of the find-path problem using the "configuration space" approach. The idea is to find those parts of free space which the object at particular orientations may occupy without colliding with an obstacle. Obstacles are "expanded" by the shape of an object at a set of orientations, while the object to be moved is shrunk to a point. The shortest path for the object, including rotations, is computed as the shortest connected path through the expanded obstacles.

The shortest path through obstacles generally leads through a sequence of points which are adjacent to the expanded obstacles. If there is position error in the control of the path execution,

such points can possibly result in a collision. Brooks has recently proposed a new approach to the find-path problem based on modeling free space [1]. Brooks' solution was developed in a two dimensional plane. Brooks fit two dimensional "generalized cylinders" to the space between obstacles to obtain pathways in which the object may freely travel on a plane. The technique was extended to the third dimension by stacking planes.

2.2 The Stanford Cart and the C-MU Rover

Moravec [11] developed a navigation system based on sensory signals using the Stanford cart. This cart sensed its environment using a set of 9 stereo images obtained from a sliding camera. A set of candidate points were obtained in each image with an "interest" operator. Small local correlations were then made at multiple resolutions to arrive at a depth estimate for the points. The matched points were plotted on a two dimensional grid and then expanded to a circle. A best path from the current location to a goal was then chosen as the shortest sequence of line segments which were tangent to the circles. The cart would advance by 3 feet and then repeat the sensing and planning process. Stereo matching was also performed between the images taken at different steps to obtain confirming and additional depth information. A new vehicle, called the C-MU Rover [12], has recently been constructed by Moravec to support these techniques.

2.3 Hilare

A team under the direction of George Giralt at the LAAS laboratory in Toulouse has been investigating the design and control of mobile robots since 1977. They have developed a mobile robot named Hilare. Chatila developed a navigation system for Hilare which is based on dividing a pre-learned floor plan into convex regions [2]. Convex regions were formed by connecting nearest vertices to form areas called C-Cells. Laumond, at the LAAS in Toulouse, extended this idea by developing hierarchies of C-Cells to represent rooms and parts of a known domain [8].

2.4 Comment

A few other efforts towards developing autonomous mobile robots have also been reported. In many cases the efforts focus on engineering problems and pay little attention to the issues of world modeling or path planning [13]. Other groups have become bogged down on the vision problem, often spending their efforts on general solutions to the problems of low level vision. We believe that the most important problems to be addressed now are sensor interpretation, navigation, and system organization. Toward this end, we have developed a computational paradigm for intelligent robotic systems. This computational paradigm provides a framework for the processes involved in sensor interpretation, path planning, and path execution.

3 The Computational Paradigm

In common usage, a paradigm is an example which serves as a model. With respect to a science, the notion of paradigm has come to represent a more specialized concept. It is a concrete puzzle solution which serves as a model for the solution of other puzzles in the science [7]. With respect to an information processing system, a computational paradigm is a framework of data structures and

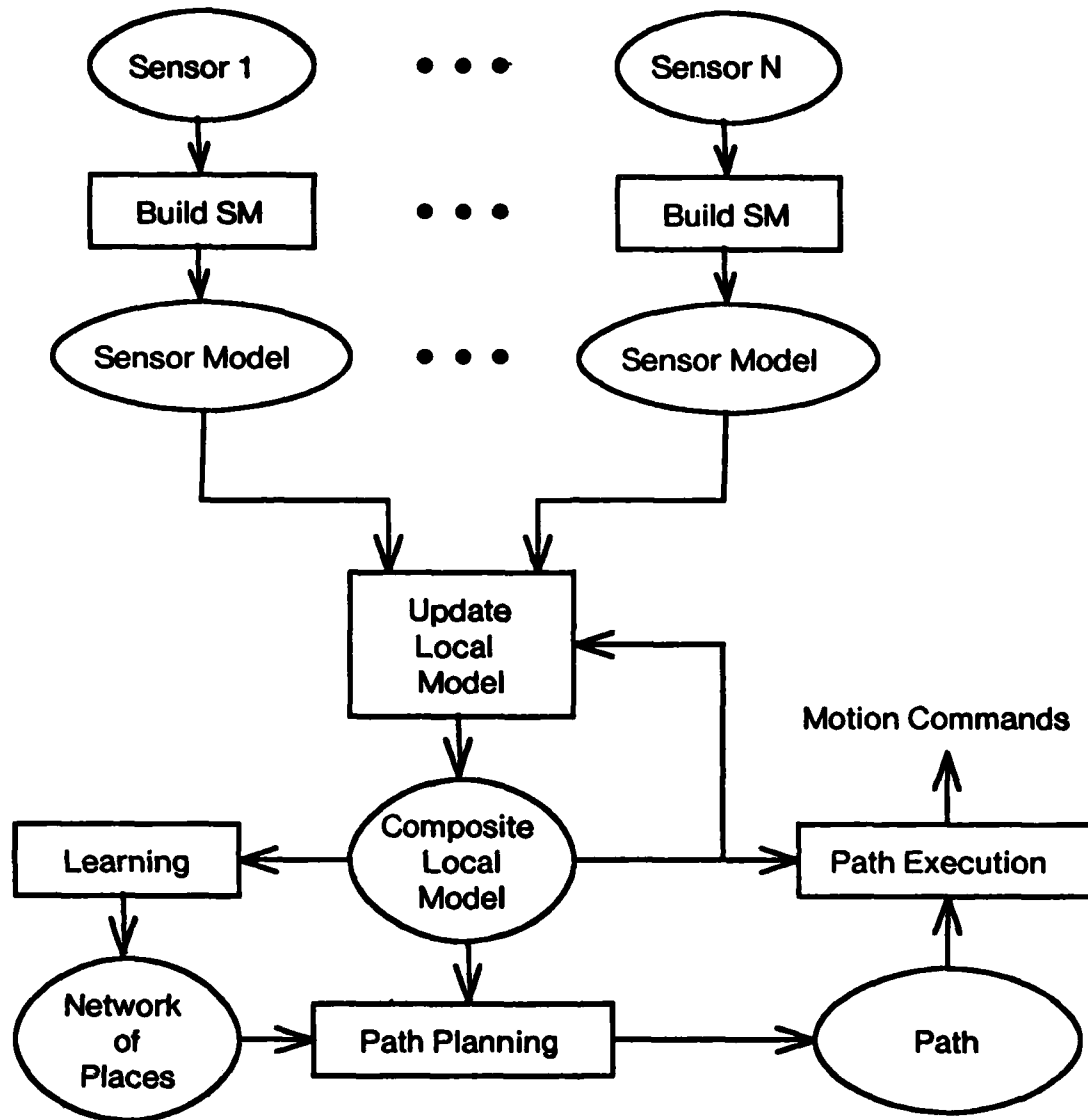


Figure 1: Framework for an Intelligent Navigation System

processes which perform some task. The navigation system of the IMP is based on the computational paradigm illustrated in figure 1. The elements of this framework are isomorphic to the elements of a computational paradigm employed by the author for dynamic three-dimensional scene analysis [5]. These systems are similar to the framework for vision proposed by David Marr [10] and others.

The navigation system of the IMP is based on maintaining a dynamic internal model of the local environment of the robot. An inexpensive rotating depth sensor continuously provides information about the external world. Differences between the sensor information and the internal model are used to indicate errors in the estimate of the position of the robot. The information from the sensor is then used to update the state of the internal model. This internal model also plays a crucial role in integrating sensor information and in providing reliable information for path planning, obstacle avoidance, learning, and path execution.

3.1 The Composite Local Model

At the core of this computational paradigm is a dynamic model of the surfaces and obstacles in the immediate environment of the IMP. This model is called "The Composite Local Model". "Local" refers to the fact that only information in the local environment of the robot is represented. "Composite" refers to the fact that this model is composed of information obtained over time from multiple sensors and from many views.

The Composite Local Model plays two fundamental roles in this computational framework.

1. It is the structure in which potentially conflicting information from diverse sensors is integrated with recently observed information and information recalled from long term storage (in the case of the IMP: the global model).
2. It is the structure on which processes for local path planning, path execution, learning, object tracking, object recognition, and other "higher level" processes are based.

Because of the nature of the navigation task and the sensors that are employed, the Composite Local Model in the IMP is implemented with a relatively simple 2-D representation. The IMP models the world and plans paths in a 2-D "flat-land" universe. Surfaces and obstacles are represented as connected sequences of line segments. Thus a table and a wall have the same structure; both appear as a barrier with an infinite (or unknown) extent in vertical dimension. The Composite Local Model must include the ability to represent the uncertainty of information. In the IMP, this ability is provided by a state transition mechanism. The line segments which compose the composite local model include a "state" attribute which represents both their source and varying degrees of uncertainty. Consistent line segments are reinforced and extended while inconsistent line segments are decayed and eventually removed from the model. This representation and the state transition mechanism is described in [4].

3.2 The Sensor Models

Sensors typically produce large amounts of information. Before the information from a sensor can be integrated into the Composite Local Model, surface information must be abstracted from it. This abstraction is performed by the module labeled BuildSM which produces a structure called the Sensor Model. The Sensor Model may be viewed as a form of "Logical Sensor" which provides the sensor information in a standard form which may be integrated into the Composite Local Model.

In the first version of the IMP, the sensors are a set of contact sensors on a skirt and the rotating sonar sensor. In each case, BuildSM abstracts information in the form of line segments representing obstacles in the real world.

3.3 Update Local Model

The module labeled Update Local Model integrates the information from the Sensor Models with the current Composite Local Model. This module consists of two parts. The first part is a matching process which establishes the correspondence between the segments in the Sensor Models and the Composite Local Model as each line is produced by BuildSM. One of the side effects of this

correspondence matching is an average error vector for the orientation and the position of the robot. This error vector tells the difference between the IMP's estimated orientation and position and its actual orientation and position. Special procedures also exist for detecting and tracking moving objects.

The second part is an integration step in which the position, size, connectivity and confidence of the segments in the Composite Local Model are adjusted to reflect the results of correspondence matching. This second stage also removes segments for which the confidence is low or for which the distance is too far. The process does not remove nearby surfaces which are not currently visible.

The problems of reconciling conflicting information and of representing uncertain information in the Composite Local Model involve interesting scientific issues. These problems are intimately related to the representation of the Composite Local Model. The techniques for representing and maintaining the Composite Local Model in the navigation system of the IMP are described in more detail in [4].

4 Navigation

The problems of path planning and path execution divide neatly into a global problem and a local problem. The global path planning process produces a path of goal points which may be traversed by a sequence of straight line motions. A global path is then executed as a sequence of local straight line movements toward these goal points. A local straight line path is executed by a process which monitors the position of the robot to assure that it remains on the desired straight line within a tolerance. This process also monitors the local model to assure that no unexpected obstacle blocks the path. Local path planning involves planning paths around unexpected obstacles. Global path planning and execution are based on information stored in the learned model of the world, whereas local path planning and execution are based on the information in the composite local model.

4.1 Navigation Modes

There are three modes in which the IMP may travel:

- | | |
|----------------|--|
| Learn Mode | Limited exploratory movements or user specified movements in an unfamiliar environment with the purpose of learning the environment. |
| Manual Mode | User specified movements in either a familiar or unfamiliar environment. |
| Automatic Mode | Autonomous movement to a goal point in a familiar environment. |

Learn mode is designed to permit the IMP to learn the network of places and the Global Model. Automatic mode is designed to permit the robot to execute navigation tasks in a learned environment. Manual mode is a default mode in which the user may drive the robot in an arbitrary manner without the restrictions required by learn mode or automatic mode.

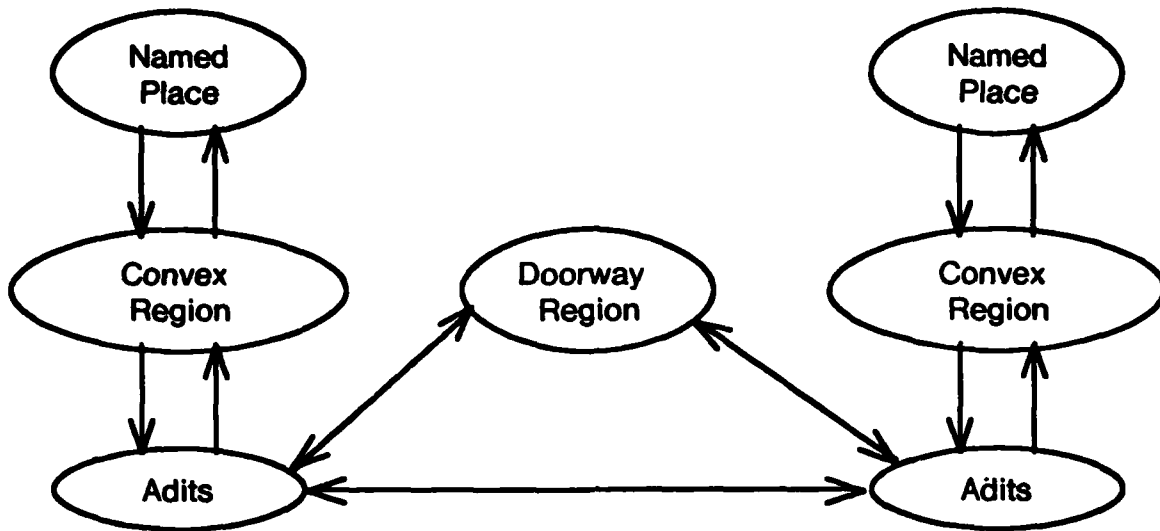


Figure 2: The Components of the Network of Places

4.2 The Network of Places and the Global Model

The known universe for the IMP is represented in two related data structures: the "global model" and the "network of places". Both of these data structures are acquired by the IMP in learn mode.

The global model is the collection of line segments observed by the composite local model while making a tour of the house in learn mode. The reinforcement process of the composite local model causes inconsistent or erroneous line segments to decay away, so that at the global model contains a clean and consistent description of the learned domain. The global model permits the IMP to recall the surfaces which it should observe at any location in the known world.

The network of places is a three level structure which serves as a basis for global path planning. This structure is represented in figure 2. The model is based on dividing the free space in the global model into convex regions. An example of the convex regions for a floor plan is shown in figure 3. A convex region has the property that any two points within the region may be connected by a straight line that remains entirely within the region. Thus a mobile robot may travel between any two points within a convex region by a single straight line motion.

Convex regions are constructed with an algorithm which is designed to maximize the area of the largest convex region [6]. The center of each boundary with an adjacent convex region is marked as a doorway. Convex regions are then shrunk by the diameter of the robot to represent the free space in which the robot may travel. Doorways are represented by a special doorway region and a pair of adits. Each adit is connected to the adit on the other side of the doorway and to all other adits within its convex region, as shown in figure 4. In this figure, the adits to the convex regions are illustrated with boxes. The convex regions are shrunk to prevent the system from planning a path which would take the robot too close to a physical barrier. These inner boundaries are represented by a thin line in figure 4.

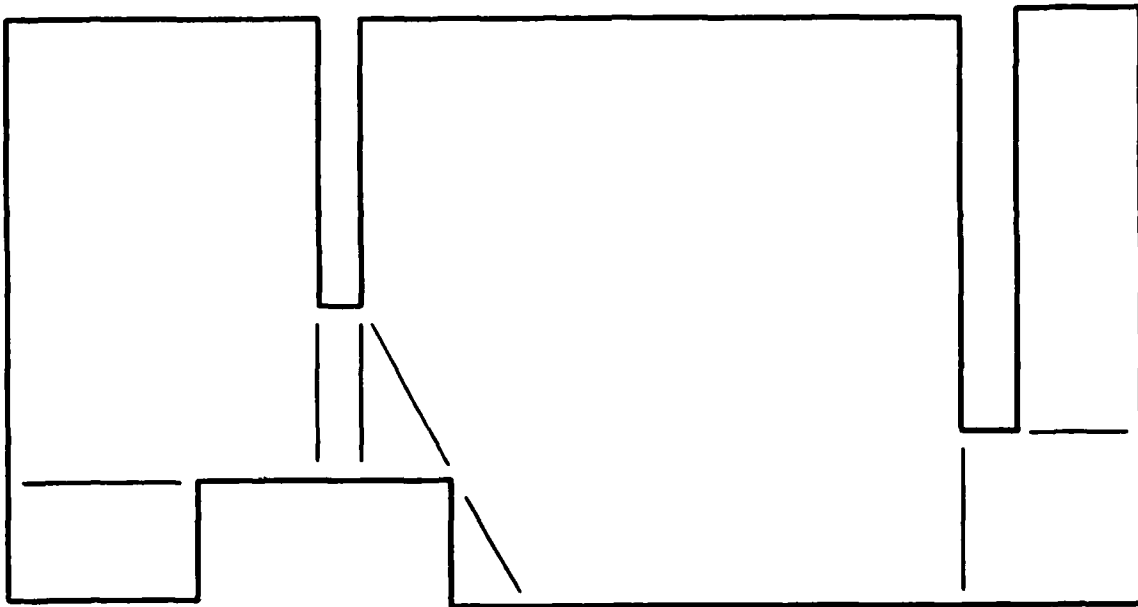


Figure 3: An example of a "largest area" convex decomposition of a floor plan.

Names are assigned to locations by the user. These named places are then accessible by their name. Each named place provides a pointer to the convex region in which it is contained. Inversely, each convex region holds a list of the named places which it contains. Each convex region also has a list pointers to the adits which it contains.

4.3 Learning the Network of Places and the Global Model

The global model and the network of places are learned by the IMP in learn mode. In this mode, the IMP explores each region, attempting to fill in gaps in its composite local model. While exploring, the IMP moves a short distance and then takes a fresh picture of the world. This is to ensure that the composite local model is as clean as possible. As each convex region is obtained, the user is given the opportunity to indicate the direction in which the IMP should continue its exploration. Named places may be added to the system by a special command in manual or automatic mode.

4.4 Global Path Planning and Execution

The network of adits provides the basis for global path planning. Global path planning begins by choosing adits for the convex regions for the goal point and for the current location of the IMP. If there are multiple adits in either region, the adit nearest to the robot (or the goal point) is chosen. The shortest path through the network of adits is then found. If the start (or the end) of this path leads

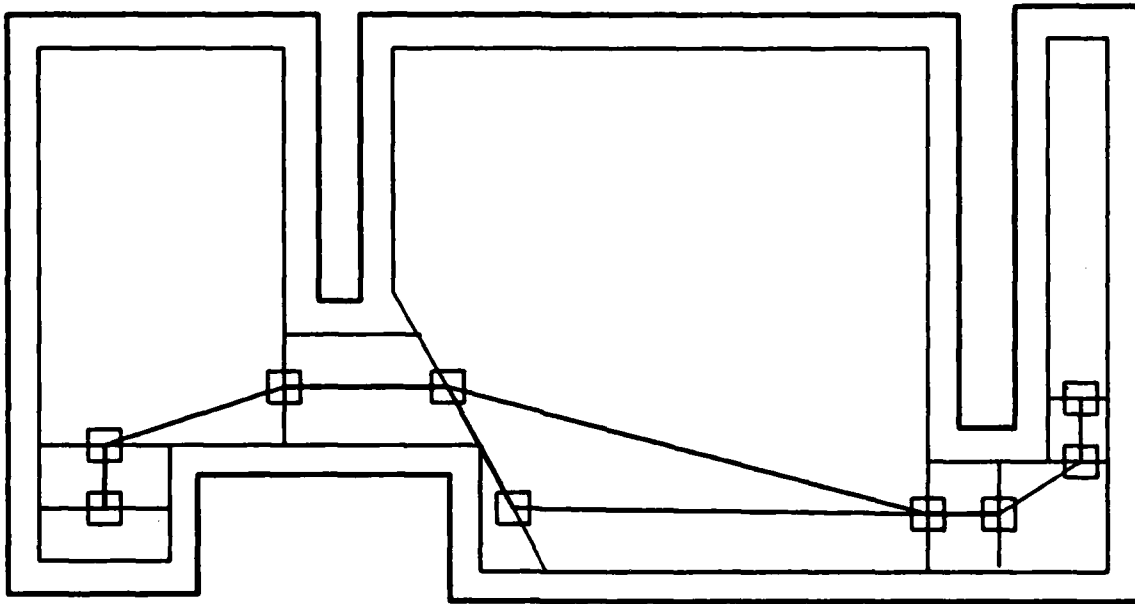


Figure 4: The set of shrunk convex regions and adits that result from the convex regions in figure 3. Adits are shown as boxes.

through two adits in the same region, the first (or last) adit is dropped from the path. Global path execution is then reduced to a three step process in which the IMP moves to the first adit in the path, moves from each adit on the path to the next, and then moves from the last adit to the goal place.

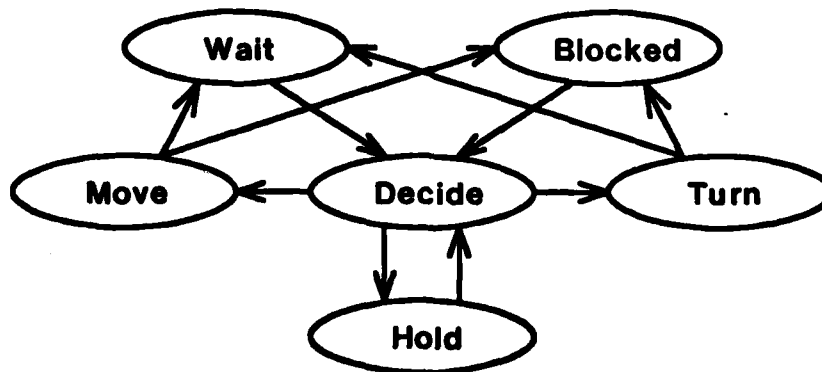


Figure 5: State Transition Diagram for Local Navigation

4.5 Local Path Execution

Straight line movement to a goal point is monitored by a relatively simple finite state process. The states of this process are the set: {Hold, Decide, Turn, Move, Wait, Blocked}. The state transition diagram for this process is shown in figure 5. The process waits in Hold state for a goal from the global path execution process. When a goal is received, the IMP enters Decide state. In Decide state it first tests the distance to the goal. If this distance is less than a tolerance, it returns to Hold state. If the distance is above the tolerance, the difference in angle between the current heading and the goal is tested. If this angle is above the minimum resolution for turning, the IMP enters Turn state. Otherwise it enters Move state.

In Turn state, the IMP turns toward the goal point until the difference between the estimated orientation and the direction to the goal point falls below the minimum turning resolution. The IMP then enters Wait state to make a complete sonar scan and verify the current estimated position. The sonar scan results in an update in the estimated position and orientation, even if there is no change in the estimated position or orientation. The call to the function "SetEstimatedPosition" signals the completion of the scan and causes a transition back to Decide state. If the IMP is not at the goal, and is turned toward the goal, control will pass from Decide state to Move state.

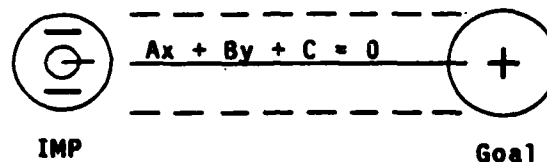


Figure 6: Legal Highway for the Path Execution

Upon entering Move state, the IMP computes the equation of the line (the path equation) from the current location to the goal point. A cyclic process is then initiated in which the system moves forward while performing the following tests as rapidly as possible.

1. Verify that the distance to the goal point is both greater than the goal tolerance and decreasing. When the distance to the goal becomes less than the tolerance, the system returns to Hold state to wait for the next goal. If the distance is ever increasing and is larger than the goal tolerance, then the system will go into Wait state to take a clean view of the world.
2. Verify that the perpendicular distance from the current estimated position to the path line segment is below a tolerance. This test is illustrated by figure 6. It is performed by evaluating the path equation using the current estimated position, yielding the perpendicular distance to the path equation. If this distance exceeds the tolerance, then the IMP will go to Wait state.
3. Verify that there is a free path to the Goal. This is done by projecting parallel line segments in the composite local model. If the path becomes blocked, the IMP will go into blocked state and signal for local path planning to avoid the obstacle.

If the IMP is in Move or Turn State, and its contact sensor is triggered, it immediately halts and

enters Blocked state. Entering Blocked state triggers the local path planning procedures to plan a path around an obstacle.

4.6 Local Path Planning

The purpose of Local Path Planning is to plan a sequence of straight line paths which will take the IMP around an unexpected obstacle. A very simple process is used, based on the segments in the composite local model. This process plans two paths, one to the left of the obstacle and one to the right. Tests are then made to see if a free path exists in the composite local model from this point to the goal, and from this point to the current position. If either test fails, the path is rejected. If both paths succeed, the shorter path is selected.

When traveling in automatic mode, the IMP also uses the previously learned description of the environment to predict the raw sensor depth reading in certain "critical" directions. These position estimation techniques are described in more detail in [Crowley84F].

5 Summary and Conclusion

This paper describes a navigation system for an intelligent mobile robot. This system is based on maintaining a dynamic model of the external world (the composite local model) using a rotating depth sensor. A side effect of maintaining this dynamic model is an error vector which is used to maintain an estimate of the robot's position as it moves. The dynamically maintained composite local model also supports the functions of path planning, learning, path execution, and object tracking.

A path planning technique has been described which is based on a pre-learned "network of places". The robot's domain is represented as a network of maximum-area convex regions. The "adits" for these convex regions serve as key points for planning paths through the known environment as sequences of straight line motions. Each straight line motion is executed by a finite state process which monitors motion using the composite local model.

These techniques yield an inexpensive navigation system which is suitable for indoor environments. This system plans and executes paths as a sequence of straight line motions. Such paths are not necessarily optimal in the sense of being shortest; they are, however, a reasonable trade-off between efficiency and safety.

References

- [1] Brooks, R. A.
Solving the Find-Path Problem by Good Representation of Free Space.
In *Proc. of the National Conference of Artificial Intelligence, AAAI-82*, pages 381-386. William Kaufman Inc., August, 1982.
- [2] Chatila, R.
Path Planning and Environmental Learning in a Mobile Robot System.
In *ECAI*. Orsay, France, August, 1982.
- [3] Crowley, J. L.
Position Estimation for an Intelligent Mobile Robot.
The C-MU Robotics Institute Annual Research Review 1(1), 1984.
- [4] Crowley, J. L.
World Modelling and Position Estimation for an Intelligent Mobile Robot.
In *Seventh International Conference on Pattern Recognition*. August, 1984.
- [5] Crowley, J. L.
A Computational Paradigm for Three Dimensional Scene Analysis.
Technical Report Submitted, C-MU Robotics Institute, January, 1984.
- [6] Crowley, J. L. And R. J. Redpath.
An Algorithm for Maximum Area Convex Decomposition.
Technical Report In Preparation, C-MU Robotics Institute, 1984.
- [7] Kuhn, T. S.
The Structure of Scientific Revolutions.
The University of Chicago Press, 1962.
- [8] Laumond, J. P.
Model Structuring and Concept Recognition: Two Aspects of Learning for a Mobile Robot.
In *Proceedings of the Eighth IJCAI-83*, pages 839-841. IJCAI, William Kaufman Inc, August, 1983.
- [9] Lozano-Pérez, T.
Automatic Planning of Manipulator Transfer Movements.
IEEE Trans on Systems Man and Cybernetics SMC-11:681-698, August, 1981.
- [10] Marr, D.
Vision.
W. H. Freeman and Co., San Francisco, 1982.
- [11] Moravec, H. P.
Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.
Technical Report CMU-RI-TR-3, C-MU Robotics Institute, September, 1980.
- [12] Moravec, H. P.
The CMU Rover.
In *Proc. of the National Conference of Artificial Intelligence, AAAI-82*, pages 377-380. William Kaufman, Inc., August, 1982.

- [13] Kanayama, Y.
Concurrent Programming of Intelligent Robots.
In *Proceedings of the Eighth IJCAI-83*, pages 834-838. IJCAI , William Kaufman Inc, August,
1983.

END

FILMED

3-85

DTIC